연성 패턴 정렬 문제

서진택°(seojt@comeng.ce.kyungpook.ac.kr), 김삼묘(kims@kyungpook.ac.kr)
경북대학교 컴퓨터공학과

Flexible Pattern Alignment Problem

Jin Taek Seo, Sam Myo Kim
Dept. of Computer Eng., Kyungpook National Univ.

요 약

본 논문에서는 1차원 스트링과 2차원 텍스트를 유동적으로 정렬하는 소위 1-2차 연성 정렬 문제를 정의하고, 이 문제를 위한 동적 알고리즘을 제시하고, 응용 예를 보인다. 문제의 패턴은 그 길이가 주어져 있지만 그 형체가 유연성을 갖고 있어 변형될 수 있다는 점이 지금까지 연구되어온 패턴 매칭 문제와 다르다.

1. 서론

스트링 매칭(matching)과 정렬(alignment) 문제 및 이의 여러 가지 변형된 문제들이 많이 연구되었으며, 최근 이는 문제에 대한 알고리즘들이 인터넷 검색 뿐 아니라 분자생물학 분야에서도 다양하게 응용되고 있다[1]. 본 논문은 유연한(flexible) 일차원 패턴을 2차원 텍스트 상에서 찾는 정렬 문제, 소위 1-2차 연성 정렬 문제(1-2 dimensional flexible alignment problem)를 정의하고 이를 위한 알고리즘을 제시한다. 차원이 같은 스트링 사이의 정확 매칭(exact matching) 및 비정확 매칭(inexact matching) 문제는 많은 연구가 있었다[3,4,6,7]. 이들 문제에서 주어진 패턴은 그 형태가 고정되어 있어 변형될 수 없다는 조건이 전제되어 있다 반면 본 논문이 연구한 매칭 문제의 패턴은 그 길이가 주어져 있지만 그 형체가 유연성을 갖고 있어 변형될 수 있다는 점이 다르다 이러한 매칭 문제는 의학 및 비파괴 구조물 영상 분석 등에 이용할 수 있을 것이다

본 논문의 구성은 다음과 같다. 2절에서는 1-2차 연성 정렬 문제를 정의하고, 3절에서는 정의한 문제를 위한 동적 알고리즘을 제시한다 다음 4절에서는 3절에서 개발한 알고리즘을 비정확 2차원 패턴 매칭 문제에 적용하는 예를 제시한 후, 5절에서 결론을 맺는다

2. 1-2차 연성 정렬 문제

이 절에서는 우리가 연구할 1-2차 연성 정렬 문제를 정의한다. 이에 앞서 편의상 2개의 스트링을 전역 정렬(global alignment)하는 문제[1]을 정의하자.

스트링 S_1과 S_2에 사용된 알파벳(alphabet)의 집합을 Σ라하고, Σ에 공백(_)이 추가된 것을 Σ'이라 하자. Σ'의 모든 문자 쌍에 대한 정렬 값은 점수 행렬(scoring matrix) s로 나타내기로 한다 따라서 x가 y에 정렬되었을 때의 값은 s(x, y)이다

정의 1. 두 스트링 S_1과 S_2에 공백을 삽입하여 정렬한 결과를 A라 하고 A 상의 각 스트링을 S_1'과 S_2'이라 하자(이때 $|S_1'|=|S_2'|$). 정렬 A의 길이를 l이라 하면, 정렬의 값은 다음과 같다.

$$\sum_{i=1}^{l} s(S_1'(i), S_2'(i)).$$

예를 들면, S_1=cacdbd, S_2=cabbdb, Σ={a, b, c, d}라 하고, 점수 행렬이 다음과 같이 주어졌다고 하자.

s	a	b	c	d	_
a	1	-1	-2	0	-1
b		3	-2	-1	0
c			0	-4	-2
d				3	-1
_					0

그러면 다음과 같은 정렬에 대한 값은 0+1-2+0+3+3-1=4 이다.

c a c _ d b d
c a b b d b _

정의 2. 알파벳 Σ'에 대해 점수 행렬 s가 주어졌을 때, S_1과 S_2의 최적 정렬 값(optimal alignment value of S_1 and S_2)은 정렬의 값을 최대화하는 S_1과 S_2의 정렬 A의 값이다

두 스트링 S_1, S_2에 대하여, 각각의 접두스트링(prefix) $S_1[1..i]$와 $S_2[1..j]$의 최적 정렬의 값을 V(i, j)로 표기하기로 하자. 두 스트링 S_1과 S_2 최적 정렬은 다음과 같은 점화식으로 표현할 수 있다[1].

$$V(0,j) = \sum_{1 \le k \le j} s(\_, S_2(k))$$

$$V(i,0) = \sum_{1 \le k \le i} s(S_1(k), \_)$$

$$V(i,j) = \max[V(i-1,j-1)+s(S_1(i),S_2(j)),\ V(i-1,j)+s(S_1(i),\_),\ V(i,j-1)+s(\_,S_2(j))] \quad \text{for } i,j>0$$

S_1과 S_2 각각의 길이를 n, m이라 하면, 최적 정렬값은 V(n, m) 상에 주어진다. 위의 최적 전역 정렬 문제는 동적 프로그래밍에 의하여 O(nm)에 풀 수 있다[1].

정의 3. 2차원 텍스트 T[1..n][1..m] 상의 임의의 점 (r, c)가 주어졌을 때, $r' \le r$을 만족하는 점 (r', c')에서 시작하여 행(x 축)으로 단조 증가(monotonic increasing)하여 (r, c)에서 끝나는 일련의 문자열을 서명 (r, c)라 하고, σ(r, c)로 나타낸다(그림 1 참조).

주어진 점 (r, c)에 대해서 많은 서명 σ(r, c)가 존재한다.

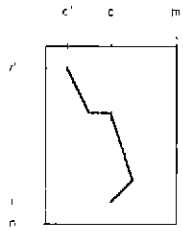

그림 1 서명 (r, c)

1-2차 연성 정렬 문제를 다음과 같이 정의한다.

정의 4. 패턴 P[1..p]와 2차원 텍스트 T[1..n][1..m]이 주어졌을 때, P의 부스트링(substring)을 α, T의 서명을 σ라 하자. P와 T로 구성할 수 있는 모든 부스트링, 서명 쌍들의 최적 정렬 값이 최대인 α와 σ를 찾는 문제를 1-2차 연성 정렬 문제라 한다.

1-2차 연성 정렬 문제를 풀이하기 위해 어느 하나가 빈 문자열(empty string)인 경우 최적 정렬의 값은 0이라고 정의한다(이러한 가정은 연성 정렬 문제에서 α와 σ가 빈 문자열이 되는 것을 허용한다). 먼저 1-2차 접미 연성 정렬 문제를 고려한다.

패턴 P[1..p]와 텍스트 T[1..n][1..m]에 대하여, $i \le p$, $r \le n$과 $c \le m$인 인덱스 3쌍(triple) (i, r, c)가 주어졌을 때, 1-2차 접미 연성 정렬 문제는 P[1..i]의 접미스트링(suffix) α와 서명 σ(r, c)의 최적 정렬 값을 찾는 문제이다. 주어진 인덱스 i, r, c에 대해 최적 접미 연성 정렬의 값을 w(i, r, c)로 나타낸다.

아래의 정리는 연성 정렬 문제와 접미 연성 정렬 문제의 관계를 보여준다. w*는 1-2차 연성 정렬 문제의 값이다(증명은 지면 관계상 생략).

정리 1. $w^* = \max\{ w(i, r, c) \mid i \le p, r \le n, c \le m \}$

3. 알고리즘(algorithm)

이 절에서는 w*를 구하는 점화식을 제시한다. 편의를 위해 s(p[i], T[r][c])를 s(i, r, c)라 표기하기로 하자. 공백의 삽입(insert)과 삭제(delete)는 -1점을 준다. 빈 문자열과의 정렬 값은 0이므로, 모든 i, r, c에 대해 w(i, 0, 0)=0, w(0, r, c)=0이다.

오른쪽(열의 증가 방향)으로 진행하는 x축, 아래쪽(행의 증가 방향)으로 증가하는 y축, 그리고 들어가는 방향으로 진행하는 z축을 가진 3차원 좌표 시스템에서, 2차원 텍스트 배열 T의 행(row)을 z축에 배치하고, 열(column)을 x축에 배치하고, 패턴 P를 y축에 배치한다.

먼저 r줄에서 x축으로 단조 증가하는 서명만 고려한다면(그것을 $w_{forward}$라 하자), 이에 대한 정렬 값은 다음과 같은 점화식으로 표현할 수 있다.

$$w_{forward}(i,r,c) = \max[0,\ w(i-1,r,c)-1, w(i-1,r,c-1)+s(i,r,c), w(i,r,c-1)-1,\ w(i-1,r-1,c-1)+s(i,r,c),\ w(i-1,r-1,c)+s(i,r,c),\ w(i-1,r-1,c+1)+s(i,r,c),\ w(i,r-1,c-1)-1, w(i,r-1,c)-1,\ w(i,r-1,c+1)-1]$$

하지만, r줄에 대해, x축으로 단조 감소(monotonic decreasing)하는 서명도 고려해야 하므로, 최종 점화식은 다음과 같다.

$$w(i,r,c) = \max[w_{forward}(i,r,c),\ w(i-1,r,c)-1, \sout{w(i-1,r,c+1)+s(i,r,c), w(i,r,c+1)-1},\ w(i-1,r-1,c-1)+s(i,r,c),\ w(i-1,r-1,c)+s(i,r,c),\ w(i-1,r-1,c+1)+s(i,r,c),\ w(i,r-1,c-1)-1, w(i,r-1,c)-1,\ w(i,r-1,c+1)-1]$$

위 점화식 중 밑줄 친 부분은 $w_{forward}$를 계산할 때 고려하지 않았던, x축으로 단조 감소하는 서명을 고려한 것이다. 또한, 전체 서명은 y축으로의 단조 증가만 고려하면 되므로, w(i, r, c)에 $w_{forward}$(i, r, c) 항을 포함해야 한다.

위의 점화식에 근거하여 동적 알고리즘을 쉽게 구현할 수 있다. 이 알고리즘에 의하여 계산된 동적 프로그래밍 큐브에서 제일 큰 값이 우리가 원하는 1-2차 연성 정렬 값이다.

4. 비정확 2차원 패턴 매칭 문제(Two-dimensional inexact matching)

이 절에서는 1-2차 연성 정렬 알고리즘의 한 응용 예로, 비정확 2차원 패턴 매칭 문제에 대한 알고리즘을 제시하고자 한다. 그림 2의 (a)같은 4×10 이진 패턴을 (b)의 16×12 이진 이미지에서 찾는 예를 생각해 보자.

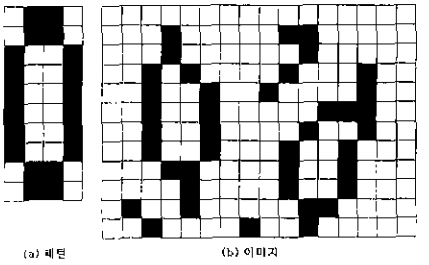

그림 2. 패턴과 이미지

패턴의 크기가 q×r이라면, 먼저 q×1 크기의 윈도우를 패턴과 이미지 위에서 이동함으로써 얻어지는 2진열을 부호화함으로써 패턴과 이미지를 변환한다. 예에서는 4×1 크기의 윈도우를 사용하여, 패턴을 길이가 10인 1차원 스트링 6699999966으로, 이미지를 아래와 그림과 같은 13×12 이차원 배열로 변환할 수 있다

0000000000000
1248001360800
1248000124800
25A4801248124
2492492480124
24924800137EC
2492480124924
2492481249248
136C801249248
0124801249248
4924800136C80
2480124924800

그림 3 부호화(encoding)된 이미지

이제 이미지에서 패턴을 찾기 위해, 서명 정렬 알고리즘을 적용하여, 패턴과 전역 정렬되는 최적 서명을 찾으면 된다

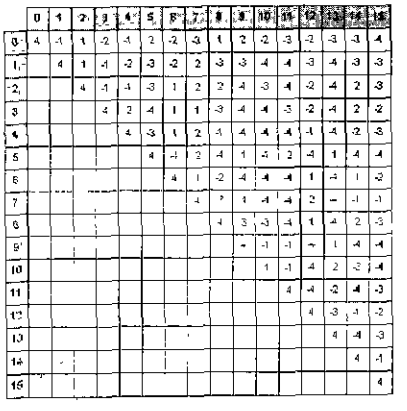

그림 4. 점수 행렬

점수 행렬은 문제에 따라 다를 수 있다. 예를 들면, 점수 행렬을 구할 때, 부호화된 이진 패턴의 모양을 고려하여, 0011과 0110은 1번의 쉬프트에 의해 패턴이 일치하므로 s(0011, 0110)=-1로 정하고, 0011과 1100은 2번의 쉬프트에 의해 패턴이 일치하므로, s(0011, 1100)=-2로 정할 수 있다. 이러한 방법으로 정의된 점수 행렬을 그림 4에 나타내었다.

예에서, 서명 정렬 V(P, σ(10,4)) = 4499996H = 15, V(P, σ(12,10)) = 44E99996H = 12 로 찾고자 하는 패턴이 코딩된 이미지 상의 점 (10, 4)와 (12, 10)에서 얻을 수 있음을 나타낸다. 알고리즘 구현 시, 큐브 상에 역포인터(back pointer)를 유지함으로써, 패턴과 정렬된 이미지의 요소를 찾아 낼 수 있다(알고리즘 구현 생략).

5. 결론

본 논문은 1-2차 연성 정렬 문제를 정의하고, 동적 프로그래밍을 이용한 알고리즘을 제시하였다. 또한 1-2차 연성 정렬을 이용하여 비정확 2차원 패턴 매칭 문제의 해를 구하는 방안을 제시하였다.

우리가 제시한 문제에서, 패턴은 그 길이가 주어져 있지만 그 형체가 유연성을 갖고 있어 변형될 수 있다는 점이 지금까지 연구되어온 패턴 매칭 문제와 다르다. 이러한 매칭 문제는 의학 및 비파괴 구조물 영상 분석 등에 이용할 수 있을 것이다.

향후 연구 과제로 서명의 형태에 제한 조건을 두는 문제도 고려할 수 있으며, 2-2차 연성 정렬 문제로 확장 연구할 필요가 있다.

참고문헌

[1] Dan Gusfield, *"Algorithms on Strings, Trees and Sequences,"* CAMBRIDGE UNIVERSITY PRESS, 1997.

[2] Graham A Stephen, *"String Search,"* School of Electronics Engineering Science University College of North Wales, October 1992

[3] Seiichi Uchida and Hiroaki Sakoe, *"A Monotonic and Continuous Two-Dimensional Warping Based on Dynamic Programming,"* IEEE, 1998.

[4] E. Levin and R. Pieraccini, *"Dynamic Planar Warping for Optical Character Recognition,"* Proc ICASSP, pages III 149-152, 1992

[5] Eric Sven Ristad, Peter N Yianilos, *"Learning String Edit Distance,"* IEEE Transaction, 1998

[6] Z Galil and K Park., *"Alphabet-independent two-dimensional witness computations"* SIAM J Comput., 25:907-35, 1996.

[7] Amihood Amir, Martin Farach, *"Efficient 2-Dimensional Approximate Matching of Half-Rectangular Figures"* Academic Press, 1995